

並列 Fast Level Set Method による実物体の高速な 3 次元形状復元

Real-time 3D Shape Reconstruction of human bodies by Parallel Fast Level Set Method

○ 学 岩下 友美 (九州大) 正 倉爪 亮 (九州大) 正 原 健二 (九州大) 正 長谷川 勉 (九州大)

Yumi Iwashita[†] Ryo Kurazume[†] Kenji Hara[‡] and Tsutomu Hasegawa[†]

[†]Kyushu University, Graduate School of Information Science and Electrical Engineering

[‡]Kyushu University, Graduate School of Design

Various markerless motion capture systems using video cameras, which enable to recover not only motion of a target but also its appearance, have been proposed so far. However, in case that there are multiple persons in the scene at the same time, it is quite difficult to reconstruct a precise 3D model of each person separately due to the occlusion between them. To deal with these problem, we have developed a motion capture system using the Fast Level Set Method for recovering 3D models of multiple persons separately and robustly against occlusion. This paper introduces an estimation method of target's moving direction based on a property of the Fast Level Set Method. As an application of the estimation method, we propose a load balancing method based on predicted moving direction for parallel Fast Level Set Method implemented on a PC cluster. Tracking experiment of a moving target using the proposed system are successfully carried out.

Key Words: Fast level set method, 3D shape reconstruction, Pc-cluster, Load balancing

1 はじめに

複数のカメラを周囲に配置して、人間の形状や動きだけではなく“見え”も計測し再現するマーカーレスのモーションキャプチャシステムの研究が行われている。このシステムの実現方法の代表的な手法として、視体積交差法と多視点ステレオが挙げられる。視体積交差法は、任意の視点から対象を撮影したとき、その対象が 2 次元シルエットを空間に投影した錐体中 (視体積) に含まれるという制約条件に基づいている¹⁾。そのため、多数台のカメラを用いることにより詳細な 3 次元形状の復元が可能となるが、一方カメラ台数の増加に伴い計算コストが高くなるという問題がある。

一方、多視点ステレオは異なる視点から撮影された 2 次元画像上で物体表面上の点の対応付けを行い、三角測量の原理により対象物体の 3 次元形状を復元する手法である。Kanade らは 51 台のカメラをドーム上に配置して多視点で計測するシステムを開発している^{2) 3)}。しかし、モデル構築のための計算コストやデータ量が非常に大きく、人間のような複雑形状の実時間計測は困難である。

さらに、従来提案されている 3 次元形状復元手法では、そのほとんどが単一物体あるいはオクルージョンの生じない複数物体を対象とした手法であり、シーン内に複数物体が存在し物体間に相互オクルージョンが生じる場合、それぞれの物体形状を同時に復元することは困難である。

これに対し、我々はこれまでにシーン内に存在する複数対象物体の 3 次元形状をオクルージョンに頑強に復元するモーションキャプチャシステムの試作機を提案した⁴⁾。試作システムでは、移動物体追跡や 3 次元幾何モデリングなど、様々な分野で用いられる Level Set Method⁵⁾の高速化手法である Fast Level Set Method⁶⁾を、対象物体の周囲に 2 台配置したステレオカメラから取得したステレオ距離画像に適用することで、複数対象物体の 3 次元形状をオクルージョンに頑強に復元している。また、7 台の計算機からなる PC クラスタへと実装し、Fast Level Set Method 処理の並列計算により、3 次元形状の実時間復元を実現している。しかし、試作システムでは並列 Fast Level Set Method 処理に割り当てるボクセル空間を単純に分割しているため、物体の位置や形状、物体数によって特定の計算機に負荷が集中する場合があった。

そこで本報告では、Fast Level Set Method の特性に着目した動物体境界の移動方向推定法を提案し、その応用例として、PC クラスタを用いた並列 Fast Level Set Method のための境界移動方向の予測値に基づく計算負荷の均衡化手法を提案する。提案手法を用いた動物体の 3 次元形状追跡実験を行い、従来の試作シ

ステムと比較してより高速に 3 次元形状の復元が可能であることを示す。

2 Fast Level Set Method

Osher, Sethian らによって提案された Level Set Method (LSM)⁵⁾は、多くの計算量を必要とするため、リアルタイム性が要求されるアプリケーションには不向きであると考えられてきた。この問題に対し、我々は LSM の高速化手法として Fast Level Set Method (FLSM)⁶⁾を提案した。以下、これらの手法を簡単に紹介する。

2.1 Level Set Method とその高速化手法

2 次元 xy 平面上での LSM を用いた境界追跡法について説明する。本手法ではまず、平面上の各点 (x, y) において、時刻 t における補助関数 $\psi(x, y, t)$ を導入し、境界位置 $\gamma(x, y, t)$ は $\psi(x, y, t) = 0$ を満たす (x, y) で表すこととする。次に、 ψ についての差分方程式を以下のように定義する。

$$\psi_{i,j}^{n+1} = \psi_{i,j}^n - F(i, j) |\nabla \psi_{i,j}^n| \Delta t \quad (1)$$

ここで、 (i, j) は xy 平面上のグリッド座標、 Δt は時間ステップ幅である。また、 F は ψ の成長速度であり、単位時間あたりに ψ が法線方向に移動する量を示す。補助関数値が 0 のセル (zero level set)、つまり $\psi = 0$ として境界を抽出することで、位相変化に対応した境界追跡が可能となる。

式 1 を更新するには、2 次元画像上の全てのセルにおいて成長速度を決定する必要がある。拡張成長速度場を用いた LSM では、各セルの成長速度を決定するために、まず zero level set での成長速度を決定し、その他のボクセルでは最も近い zero level set のボクセルの成長速度をコピーして成長速度場を構築する。しかし、拡張成長速度場の構築処理において、各セルで現在の zero level set からの距離を得る際、各セルからの最近傍 zero level set の探索処理を行わなくてはならない。この計算コストは非常に高く、これが LSM の大きな問題となっている。

LSM の計算コストを削減するために、これまでに様々な手法が提案されており、その代表的な手法として Narrow Band Method (以下, NB) が挙げられる。一般に境界領域の追跡において、空間全体に対して補助関数を計算する必要はないことから、この手法では zero level set に近い領域 (Narrow Band) だけに処理を限定することで処理の効率化を図っている。

2.2 Fast Level Set Method

NBはLSMと比べて高速な手法であるが、依然として計算コストは高い。この問題に対し、LSMの高速化手法としてFast Level Set Method (FLSM)⁶⁾を提案した。

FLSMはFig.1 (a)に示す参照マップを用いて、最近傍探索処理をあるルールに基づく単純な数値の上書き処理に置き換えることで、高速に拡張成長速度場を構築する。この参照マップは、原点周辺にあるセルを原点からの距離に応じて分類したものである。例えば、リスト R_r は原点からの2乗距離が r であるセルの集合とする。

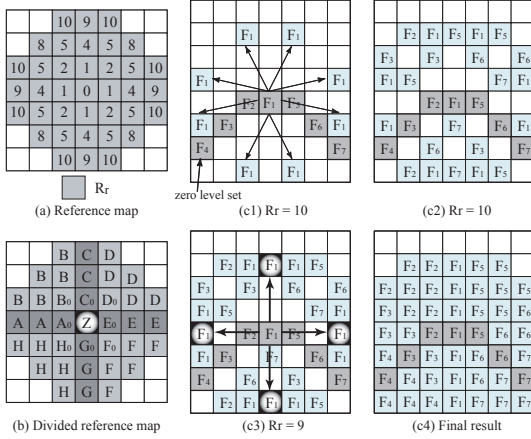


Fig.1: Reference map and the construction process of the extension velocity field.

次に、作成した参照マップを用いて拡張成長速度場を構築する。ただし、zero level setでの成長速度は予め決定されているとする。まず、リスト $R_{\delta(\delta+1)}$ を用いて、あるzero level setからの2乗距離が $\delta(\delta+1)$ であるセルを選択し、そのzero level setに格納されている成長速度の値を選択されたセルに仮登録する (Fig.1 (c1))。この処理をすべてのzero level setに対して行う (Fig.1 (c2))。次に、添字の値を1小さくして同じ処理を行い (Fig.1 (c3))、これを添字の値が0になるまで繰り返す。ただし、仮登録の際、異なる値がすでに仮登録されていた場合には、新たな値を上書きすることにする。これにより全ての処理が終了した時には、各セルには最も近いzero level setにおける成長速度の値が登録されている (Fig.1 (c4))。このように、参照マップ球の距離に応じたリストを利用することで、距離比較を行うことなく代入処理だけで拡張成長速度場を構築する。

さらにFLSMではFig.1 (b)に示すように分割した参照マップを用いることで、拡張成長速度場の構築処理において書き込む領域を限定し、さらに効率化を実現している。例えば、あるzero level set Z の左側にzero level set A_0 が隣接している場合、 Z の左側の領域 (A, B, H)には、 A_0 よりも Z に近い点は存在しない。このように、隣接するグリッドがzero level setかどうかを調べ、その位置関係によって、書き込みのオーバーラップを減らし、拡張成長速度場の高速な構築処理を可能としている。

3 境界の移動方向推定及び並列Fast Level Set Methodのための負荷分散の最適化

本章ではまずこれまでに提案した、複数対象物体の3次元形状をオクルージョンに頑強に復元するモーションスキャナシステムの試作機について概説する⁴⁾。従来、様々なオクルージョンの対応手法が提案されているが、一般に推定される物体形状は既知かつ楕円などで近似された単純な構造であり、自由形状物体のオクルージョン判定には拡張されていない⁷⁾。次にFLSMの特性に着目した動物体境界の移動方向推定手法を提案し、その応用例として、PCクラスタを用いた並列FLSMのための境界移動方向の予測値に基づく計算負荷の均衡化手法を提案する。

3.1 FLSMによる人体形状の復元と追跡

開発したモーションスキャナシステムの処理の流れを以下に示す。

1. まず内部パラメータが既知の校正済みステレオカメラ (Point-Greay社製、BumbleBee) を大きさ・形状が既知の静止物体 (例えば立方体) の周囲に設置し、物体の形状モデルとステレオ距離画像を用いてICPアルゴリズムにより外部パラメータを推定する。
2. 測定対象を内部に設置し、全てのステレオカメラを用いてステレオ距離画像を取得する。
3. 複数距離画像を一つの3次元ボクセル空間に投影し、距離データを含むボクセルを抽出する。
4. 取得したステレオ距離画像と前フレームで復元した3次元形状情報を用い、オクルージョン領域を推定する。
5. 3.で抽出されたボクセル領域に対しFLSMを適用し、距離データ点を含む最も外側のボクセル (境界ボクセル, zero level set) を追跡し、複数対象物体の分離した3次元形状を復元する。オクルージョン領域内の形状は、前時刻での境界位置を保つことで連続した滑らかな形状を保持する。
6. 2から5を繰り返す。

上記の計算は、処理の高速化を図るため、ギガビットLANの一つであるMyrinetxpで相互に接続された7台の計算機 (Pentium Xeon, 3.06 GHz) により並列処理される (Fig.2)。まずそれぞれステレオカメラが接続された2台の計算機 (PC1, PC2) により、同期した2枚のステレオ距離画像を取得する。次に得られた距離画像はFLSM計算用計算機に送られる。ただしここではボクセル空間を4つの領域に均等に分割し、それぞれの空間に1台ずつ計算機を割り当て (PC3-PC6)、4台で並列にFLSMの計算を実行した。各FLSM用計算機では、まず複数距離画像の統合を行った後、それぞれの領域に対してFLSMを適用する。最後に処理結果が表示用計算機 (PC7) に送られ、ボクセル空間全体の処理結果として統合、表示される。

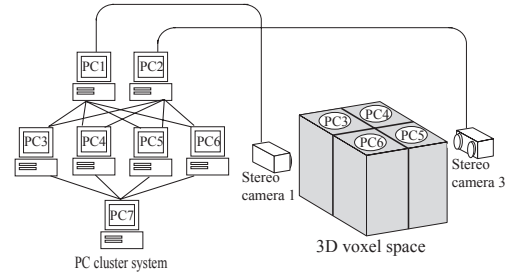


Fig.2: PC cluster system.

3.2 対象物体の移動方向の推定

次に動物体境界の移動方向の推定手法について述べる。まず、時刻 $t - \Delta t$ で構築した3次元モデルの境界ボクセル $Z(t - \Delta t, i)$ ($0 \leq i < N(t - \Delta t)$) と時刻 t で構築した3次元モデルの境界ボクセル $Z(t, j)$ ($0 \leq j < N(t)$) との対応付けを行う。ここで、 $N(t)$ は時刻 t で復元した3次元モデルの境界ボクセルの総数とする。具体的には、まず時刻 $t - \Delta t$ で復元した境界ボクセル $Z(t - \Delta t, i)$ にラベル $L(t - \Delta t, i)$ を格納し、距離場の構築処理と同様の手順でNarrow Band内のボクセルにラベル $L(t - \Delta t, i)$ をコピーする。全ての境界ボクセルのラベルをNarrow Band内のボクセルにコピーすると、Narrow Band内のボクセルには3次元形状の最近傍境界ボクセルのラベルが格納される (Fig.3(a))。次に、時刻 t で構築した3次元形状の境界ボクセル $Z(t, j)$ に格納されているラベルより、境界ボクセル $Z(t - \Delta t, i)$ と境界ボクセル $Z(t, j)$ との対応付けを行い、相対変位を表すベクトル $Link(t, j) = Z(t, j) - Z(t - \Delta t, i)$ とする。ここで、時刻 $(t - \Delta t) \sim t$ において対象物体の局所的な形状は保持され、かつ

対象物体の加速度は 0 と仮定すると, $Link(t, j)$ と境界ボクセル $Z(t, j)$ 近傍の境界面の法線ベクトル $n(t, j)$ との一致度より, 対象物体の移動方向を推定することができる (Fig.3.(b)). そこで, 時刻 t における対象物体の移動方向ベクトル $MD(t)$ は

$$MD(t) = (1.0 - \alpha) \sum_{|Link(t, j) \cdot n(t, j)| > \beta} Link(t, j) + \alpha MD(t - \Delta t) \quad (2)$$

と定義する. ただし, α, β は定数である.

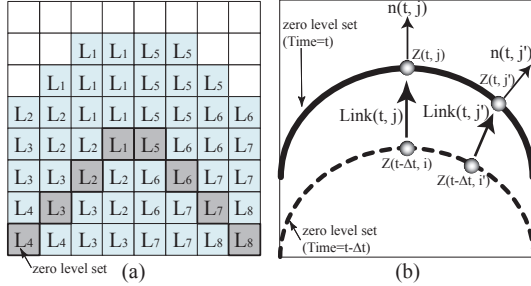


Fig.3: Moving direction estimation of moving object.

対象物体の移動方向を推定する手法として, 例えば復元した 3 次元形状の重心位置の時系列データを用いることが考えられる. しかし, 例えば腕や足などの体の一部のみを動かす場合, 全体の重心位置を用いる手法ではその部位のみの移動方向を推定することは困難である. これに対し, 提案した移動方向推定法は, 境界の局所的な移動方向の推定が可能であり, 移動部位の切り出しも期待できる.

例として半径 0.1m の球体を用いて移動方向推定のシミュレーション実験を行い, その推定性能を評価した. 実験条件は, X 軸, Y 軸, Z 軸に沿ってそれぞれ 0.01m ずつ, 合計 1.5m 動かした場合と, Y 軸を回転軸に 5 度ずつ回転した場合の 4 通りである. また, 軸方向へ移動する場合は $\alpha = 0.0, \beta = 0.98$ とし, 軸周りで回転する場合は $\alpha = 0.4, \beta = 0.98$ とした. Table1 に推定された移動方向ベクトルと実際の移動方向ベクトルとの角度の差の平均と標準偏差を示す. これより, 提案手法により物体の移動方向の推定が可能であることが確認できる. 一方, 軸方向への移動と比較し, 軸周りの回転の推定結果が悪いことがわかる. これは, 3 次元形状の境界ボクセルに格納されているラベルを Narrow Band 内にコピーする際, 異なるラベルが既に仮登録されている場合には, 新たなラベルを上書きするため, 正しいラベルが保持されない場合があるためと考えられる. この解決法として, 復元した 3 次元形状のラベルを Narrow Band 内にコピーする際に, 3 次元形状の法線方向へのコピーを優先することが挙げられる.

3.3 並列 FLSM のための負荷分散の最適化

3.1 節で示した従来のシステムは, 並列 FLSM 処理においてボクセル空間を単純に分割しているため, 物体の位置や形状, 物体数によって特定の計算機に負荷が集中する場合があった (Fig.4(a)). そこで推定した動物体の移動方向情報を用いた, PC クラスタを用いた並列 FLSM のための計算負荷の均衡化手法を提案する.

まず, 対象物体の位置と復元した 3 次元形状のボクセル数に応じて, ボクセル空間の分割面を決定する. 具体的には, まず FLSM 計算機において復元したそれぞれの 3 次元形状情報を表示用計算機へと送信する. 次に表示用計算機において対象物体の形状分布を統合し各領域内の 3 次元形状のボクセル数が等しくなるようにボクセル空間を 4 つの領域へと分割し, ボクセル空間の分割面を更新する (Fig.4(b)). 最後に, 更新したボクセル空間の分割面情報を各 FLSM 計算機へと送信し, FLSM 計算機では更新された分割面に応じて FLSM 処理領域を変更する. しかしこの手法では, 並列 FLSM 処理において更新されたボクセル空間の分割面情報は, 表示用計算機で更新されるため, 実際には前時刻以前で復元した 3 次元形状分布を用いることになる. このため, 各 FLSM 計算機間で復元する 3 次元形状のボクセル数は正確には均一化されない場合がある.

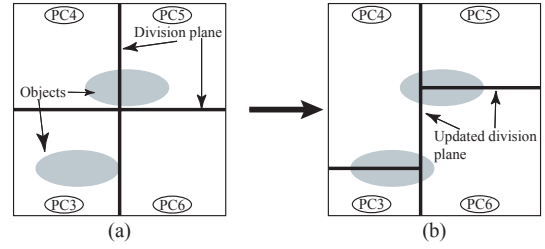


Fig.4: Adaptive division of voxel space.

そこで, 3.2 節で提案した動物体の移動方向の予測値を利用し, 並列 FLSM に割り当てるボクセル空間領域をより最適に決定する. 具体的には, 物体形状分布を用いて決定した分割点 $D(i, t) = (d_x(i, t), d_y(i, t), d_z(i, t))$ ($i = 0 \sim \text{FLSM の PC 数}$) とすると, 新たな分割点 $D_{new}(i, t) = (dn_x(i, t), dn_y(i, t), dn_z(i, t))$ を以下のように決定する.

$$D_{new}(i, t) = D(i, t) + \gamma MD(t) / |MD(t)| \quad (3)$$

ただし, γ は定数である. これにより, 各計算機で復元する 3 次元形状のボクセル数はほぼ均一となり, 並列 FLSM の負荷分散の最適化が期待される.

4 実験

提案したモーションスキャナシステムを移動する人体へ適用し, 境界移動予測値を用いた負荷均衡化手法の有効性を確認する実験を行った. 使用したボクセル空間の解像度は $100 \times 100 \times 100$ (ボクセル空間の一辺は 1.5cm) である.

まず, Fig.5(a) のように腕をあげた状態でシーン内を移動する対象物体の追跡実験を行った. ただし, 並列 FLSM に割り当てるボクセル空間領域を単純に 4 分割した場合 (Exp.A), 形状分布から分割面を決定した場合 (Exp.B), 及び対象物体の移動方向の予測値に基づいてボクセル空間を分割する場合 (Exp.C) について実験を行った. Fig.5(b) に (Exp.A) の結果, Fig.5(c) に (Exp.C) の結果をそれぞれ示す. この結果より, (Exp.A) では特定の計算機に FLSM 計算負荷が集中していたため, 腕の一部に欠損が生じているが, (Exp.C) では, 欠損を生じることなく 3 次元形状を復元している. また Table2 に (Exp.A), (Exp.B), 及び (Exp.C) 対象物体の移動方向の予測値に基づいてボクセル空間を分割する場合の計算時間の比較を行った. ただし $\alpha = 0.3, \beta = 0.9, \gamma = 4$ である. これより, (Exp.C) の物体の移動方向の予測値に基づく FLSM 計算機間の計算負荷の均衡化を行う場合, 復元する 3 次元形状のボクセル数が増加するにも関わらず, より高速に 3 次元形状の復元が可能であることがわかる.

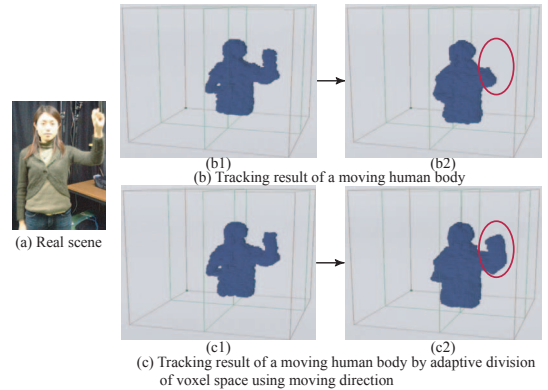


Fig.5: 3D model reconstruction using load balancing method.

さらに提案した手法に基づき, ステレオカメラ 5 台を用いて詳細な人間動作の追跡実験を行った. 実験では Fig.6 に示すように, ステレオカメラ 1~4 は側面に, またステレオカメラ 5 は天

Table 1: Accuracy of the estimated moving direction.

	Average [deg.]	Standard variation [deg.]
Translation along the x-axis	0.79	1.77
Translation along the y-axis	0.51	1.59
Translation along the z-axis	2.45	2.72
Y-axis rotation	4.67	2.45

Table 2: Computation time of the parallel FLSM.

Voxel space division	Average calculation time [msec./1voxel]	Total voxel number of the reconstructed 3D model
(Exp.A) Equal division	0.015	7638
(Exp.B) Adaptive division	0.012	9004
(Exp.C) Adaptive division using moving direction	0.010	12019

井に設置した。Fig.7(a) に対象物体の実際の様子、及び Fig.7(b) に復元した形状を示す。また生成したモデルに対し、マーチングキューブ法を適用し、その後テクスチャマッピングを行った結果を示す (Fig.7(c))。

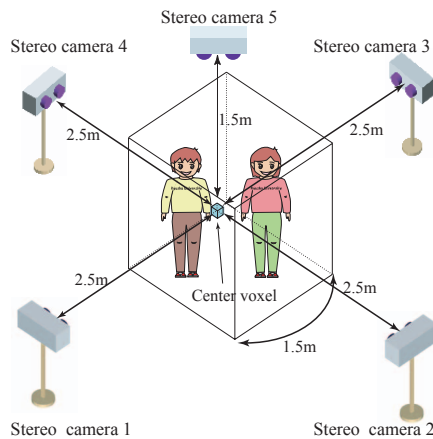


Fig.6: 3D motion scanner using multiple stereo cameras and the camera settings.

5 まとめ

本報告ではシーン内に存在する複数対象物体の3次元形状をオクルージョンに頑強に復元するモーションスキャナシステムの概要を示した。また、Fast Level Set Method の特性を用いた動物体の移動方向推定手法を提案し、その応用例として、PC クラスタを用いた並列 Fast Level Set Method のための境界移動方向の予測値に基づく計算負荷の最適化手法を提案した。動物体の追跡実験により従来システムと比較して、より高速な3次元形状の復元が可能であることを示した。今後は増設したステレオカメラを用い、シーン内に存在する複数対象物体の3次元形状をオクルージョンに頑強に実時間で復元するシステムを開発する。

- 1) W. Martin and J. Aggarwal, "Volumetric description of objects from multiple views", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.5, No.2, pp.150-158, 1983
- 2) T. Kanade, P. Rander and P. Narayanan, "Virtualized Reality: Constructing Virtual Worlds from Real Scenes",

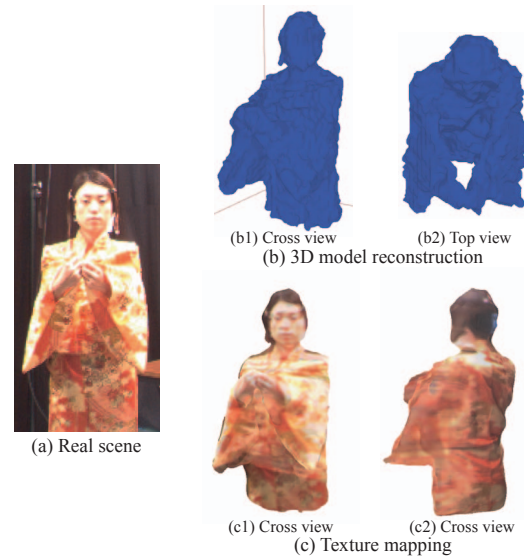


Fig.7: Precise 3D model reconstruction.

IEEE Multimedia, Immersive Telepresence, vol.4, No.1, pp.34-47, 1997

- 3) S. Vedula and S. Baker and T. Kanade, "Image-Based Spatio-Temporal Modeling and View Interpolation of Dynamic Events", *ACM Transactions on Graphics*, vol.24, No.2, April, 2005
- 4) 岩下, 倉爪, 辻, 原, 長谷川, "Fast Level Set Method を用いた複数移動物体の3次元追跡", *日本ロボット学会論文誌*, vol.23, No.7, pp.813-820, 2005
- 5) J. Sethian, "A fast marching level set method for monotonically advancing fronts", *Proceedings of the National Academy of Science*, vol.93, pp.1591-1595, 1996
- 6) 倉爪, 由井, 辻, 岩下, 原, 長谷川, "Fast Level Set Method の提案とビデオ画像の移動物体のリアルタイム追跡", *情報処理学会論文誌*, vol.44, No.8, pp.2244-2254, 2003
- 7) K. Otsuka and N. Mukawa, "Multiview occlusion analysis for tracking densely populated objects based on 2-D visual angles", *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol.1, pp.90-97, 2004